



Софийски университет „Св. Климент Охридски“  
Факултет по математика и информатика

## Домашно за поправителна сесия

курсове *Структури от данни;  
Структури от данни и програмиране*

специалности *Информатика, Информационни системи и  
Компютърни науки (1-ви поток)*

*Поправителна сесия 2021/22 г.*

### Редакции

---

02.06.2022 - Уточнение на операцията премахване при повече от едно срещане.

15.06.2022 - Уточнение за премахване - да не се добавя елемент, когато вече има наследник със същата стойност.

### Условие на задачата

---

Напишете програма, която работи с дървета съдържащи във възлите си цели числа и имащи произволна разклоненост. Числата са произволни стойности от тип `int` и могат да се повтарят в дървото. Единственото ограничение е, че един възел не може да има два преки наследника с една и съща стойност.

В тази задача можете да използвате структурите от данни и алгоритмите от стандартната библиотека.

При стартирането си, вашата програма трябва да може да въвежда команди от потребителя и да ги изпълнява. Командите, които трябва да се поддържат са описани по-долу.

Програмата трябва да може да зарежда дървета и да им съпоставя имена. Сложността по време на намирането на дърво по неговото име трябва да бъде не повече от  $O(k)$ , където  $k$  е дължината на най-дългото име на дърво заредено в програмата. Тази сложност НЕ ТРЯБВА да зависи от броя на дърветата заредени в програмата.

LOAD <name> <file>

Прочита дърво от текстов файл <file> и го съхранява в паметта, като му дава име <name>. Програмата трябва да може да зарежда произволен брой дървета. Ако се направи опит да се зареди дърво с име като вече съществуващо, да се презапише старата информация. Ако <file> не може да се отвори за четене, да се изведе съобщение за грешка и програмата да продължи своето изпълнение.

<name> може да бъде произволен низ, в който не се съдържат интервали.

Във файла дървото е описано по следния начин:

- Всеки ред представя едно ниво от дървото. Елементите на реда са разделени с един или повече интервали.
- В едно ниво братствата са разделени с вертикална черта |. Всеки ред започва и завършва с такава черта.
- Ако на дадено ниво елемент е листо, то на следващото ниво братството на неговите наследници е празно | |.
- Братства съществуват само за елементи от предното ниво (т.е. ако на ниво N има листо, то на ниво N+1 имаме празно братство за него и на всяко по-долно ниво няма следа от този елемент).

Примери за входни файлове ще намерите в края на описанието на задачата.

SAVE <name> <file>

Записва дървото с име <name> във файла <file>. Форматът да бъде този, който е описан под командата LOAD. Ако <file> вече съществува, програмата трябва да попита потребителя дали да се презапише неговото съдържание или да се откаже операцията. Ако възникне грешка (например <name> не е валидно име на дърво, <file> не може да се отвори за запис и т.н.), да се покаже подходящо съобщение за грешка и програмата да продължи своето изпълнение.

CONTAINS <tree1> <tree2>

Проверява дали дървото <tree2> се среща някъде в <tree1>. Това означава, че в <tree1> съществува поддърво, чийто корен може да се съпостави на корена на <tree2> и при премахване на елементи от това поддърво ще се получи <tree2>. Позволено е пренареждане на елементите в братство, но не и между нивата на дървото. При съвпадение на два възела трябва да се съблюдава съвпадение на стойностите им.

REMOVE <tree1> <tree2>

Премахва всяко срещане (съгласно описаното за командата CONTAINS) на <tree2> от <tree1>. Ако при премахване на срещането останат под-дървета, за които бащата на корена им е премахнат, то те да се премахнат също, но сумата от елементите им да се

съхрани като един елемент, който да се постави на мястото на корена на премахнатото дърво. Ако вече има наследник с такава стойност, то нов елемент не се добавя. Ако има множество срещания да се започне от най-ниско разположеното. Имайте предвид, че промените от едно премахване може да породят следващо срещане.

## Примери

---

### Съдържание на "C:\Sample Trees\tree1.txt"

```
| 8 |  
| 3 2 9 |  
| 2 | 8 6 | 2 | | |
| 8 6 | 11 | 12 | 8 5 |  
| 11 | 4 3 | | 11 | 6 |  
| | | | |
```

### Съдържание на "C:\Sample Trees\tree2.txt"

```
| 2 |  
| 8 6 |  
| 11 | |  
| |
```

### Команди:

```
LOAD t1 C:\Sample Trees\tree1.txt  
LOAD t2 C:\Sample Trees\tree2.txt  
REMOVE t1 t2  
SAVE t1 C:\Temp\result.txt
```

### Съдържание на "C:\Temp\result.txt"

```
| 8 |  
| 3 12 9 |  
| 7 | | 2 |  
| | 8 5 |  
| 11 | 6 |  
| | |
```

### Диаграми на дърветата от примера:

